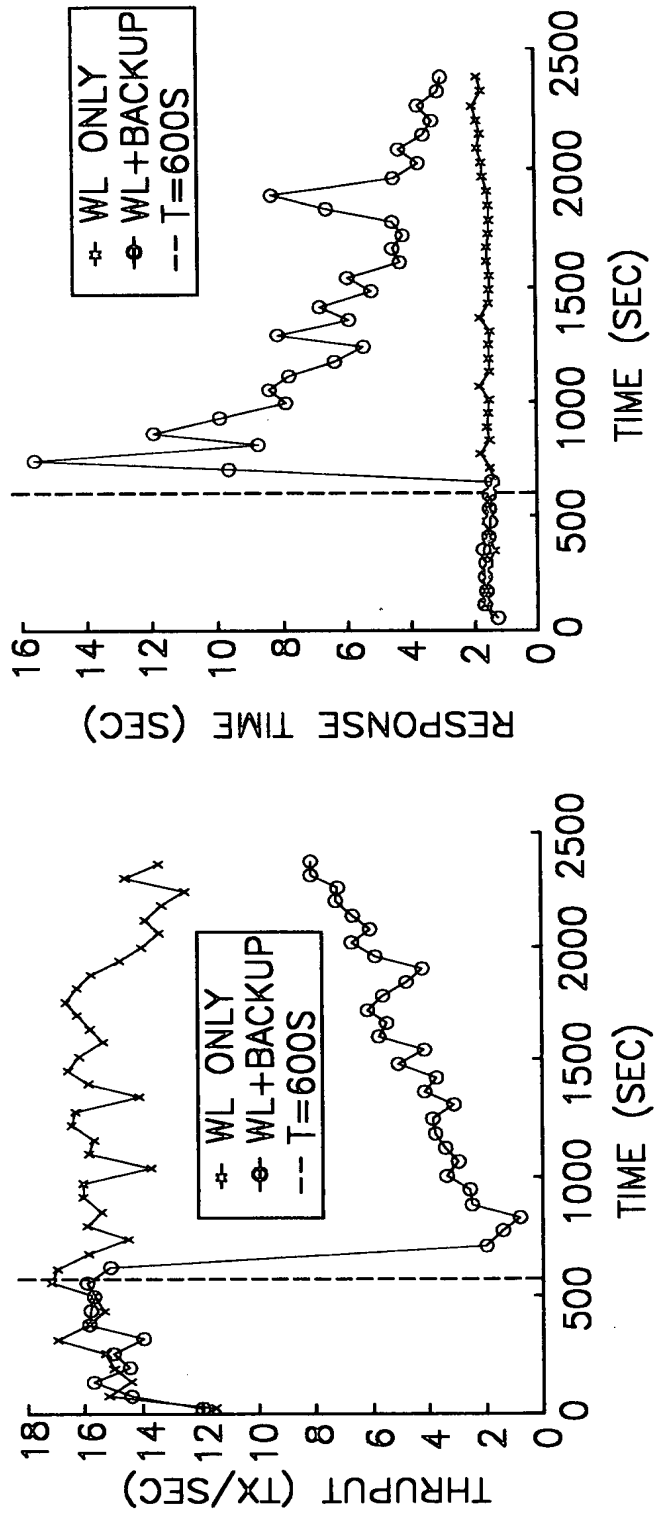

FIG. 1

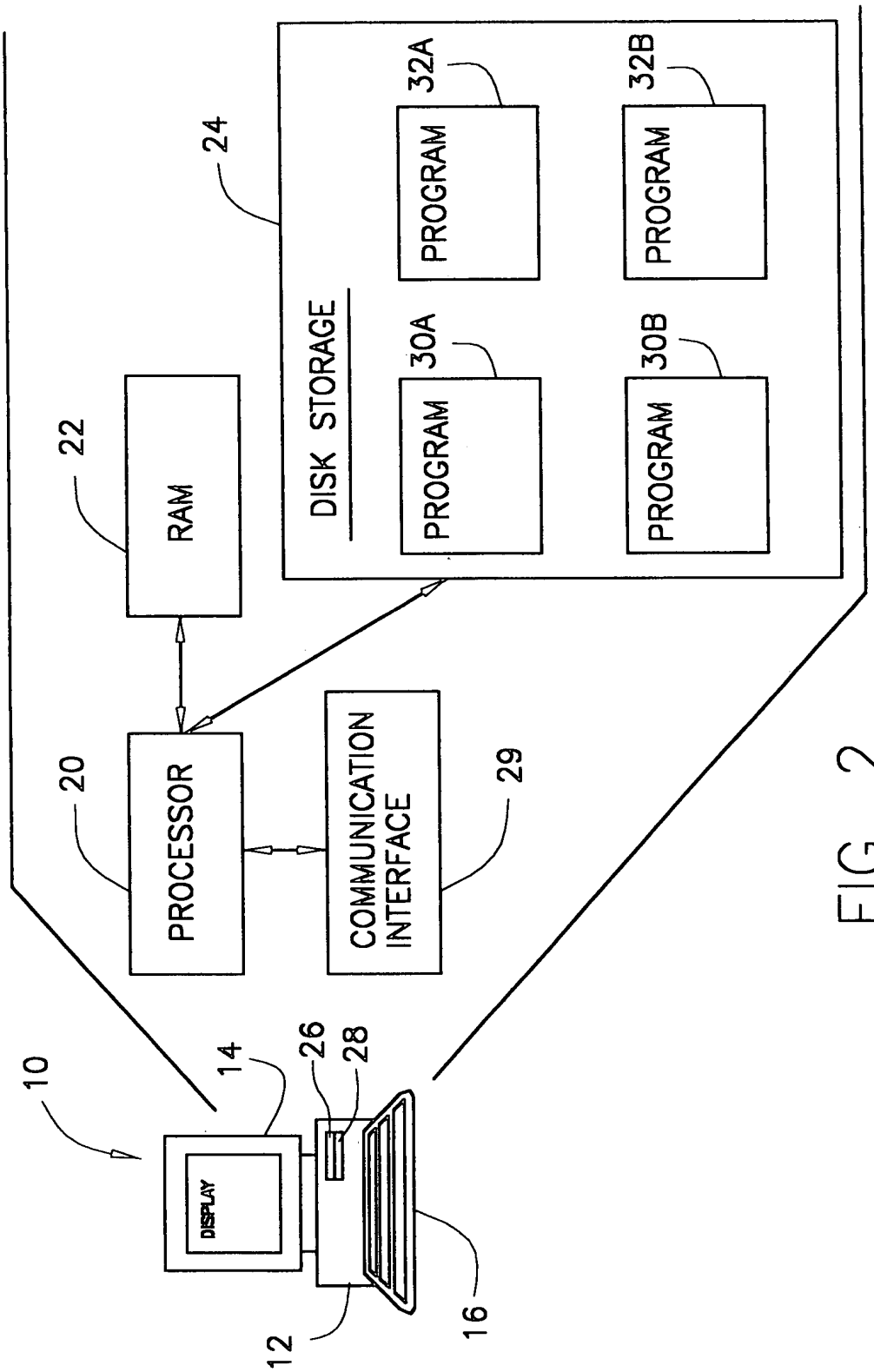
10
12
14
16
20
22
24
26
28
29
30A
30B
32A
32B
DISPLAY
PROCESSOR
RAM
DISK STORAGE
PROGRAM
PROGRAM
PROGRAM
PROGRAM
COMMUNICATION INTERFACE

FIG. 2

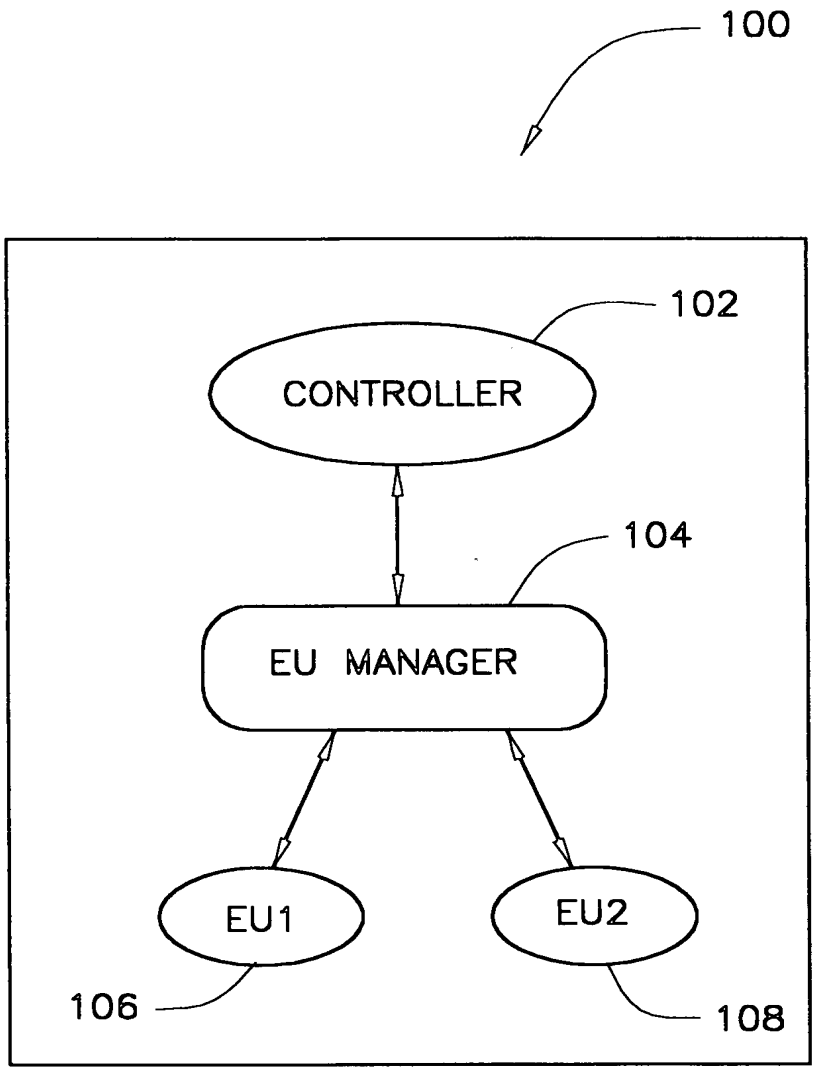

FIG. 3

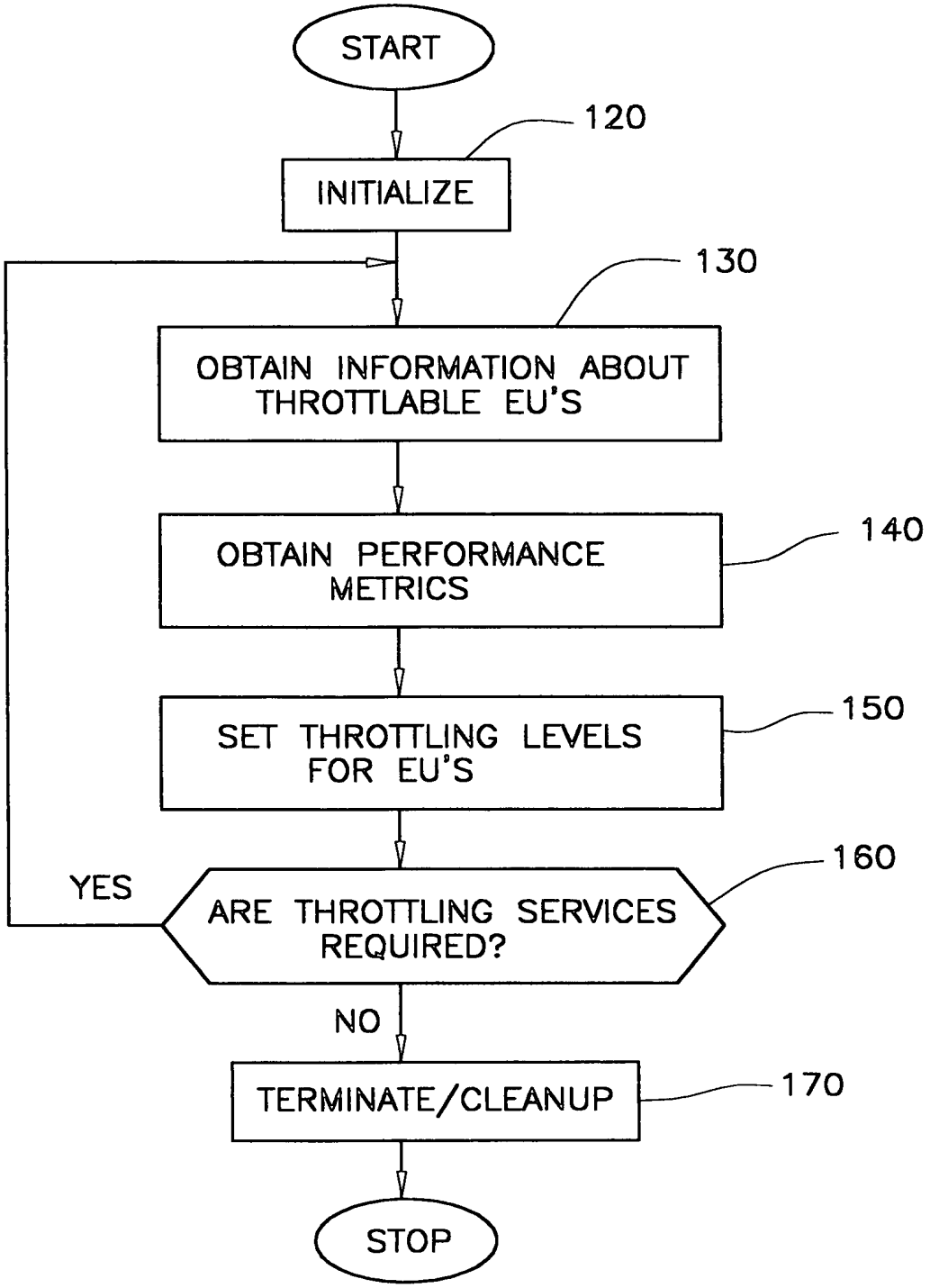
START
120
INITIALIZE
130
OBTAIN INFORMATION ABOUT THROTTLABLE EU'S
OBTAIN PERFORMANCE METRICS
140
SET THROTTLING LEVELS FOR EU'S
150
160
YES
ARE THROTTLING SERVICES REQUIRED?
NO
TERMINATE/CLEANUP
170
STOP

FIG. 4

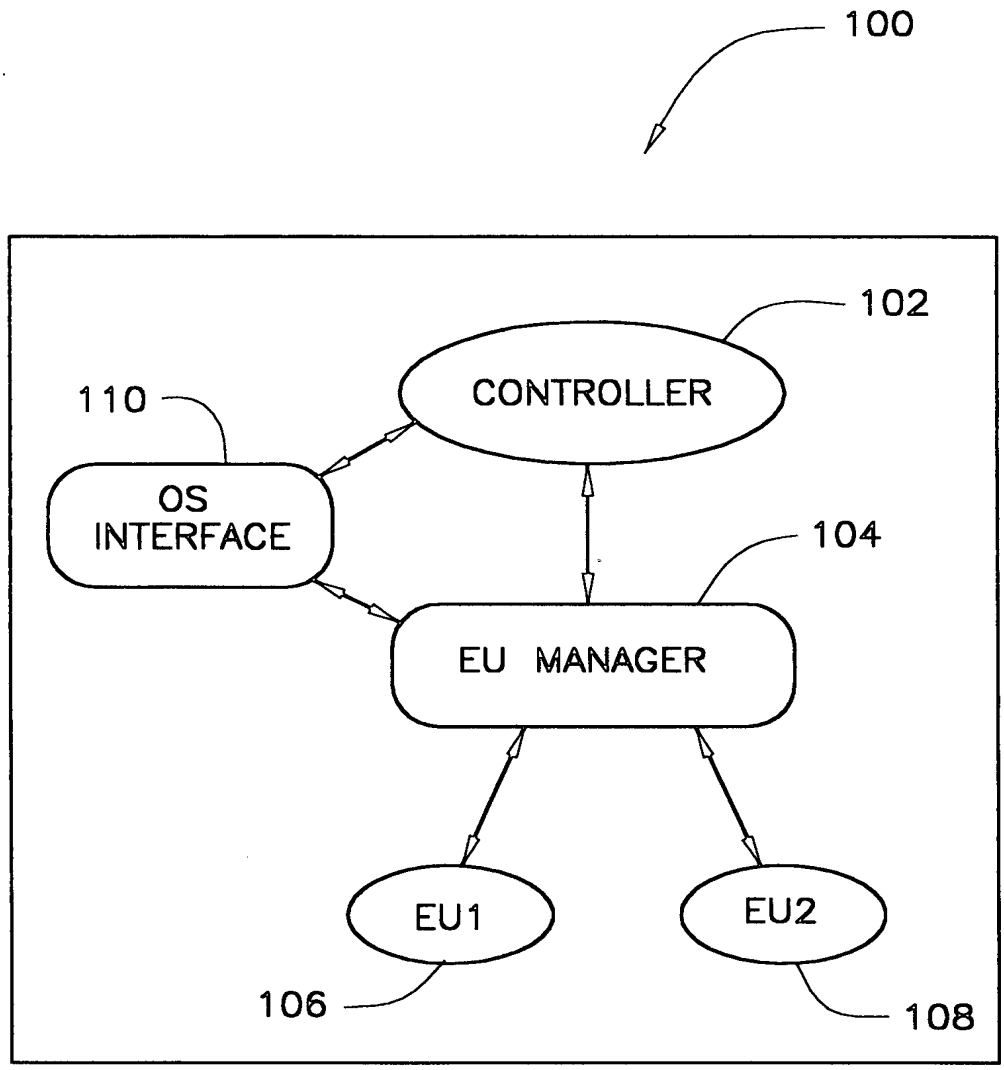
100
102
CONTROLLER
110
OS
INTERFACE
104
EU MANAGER
EU1
EU2
106
108

FIG. 5

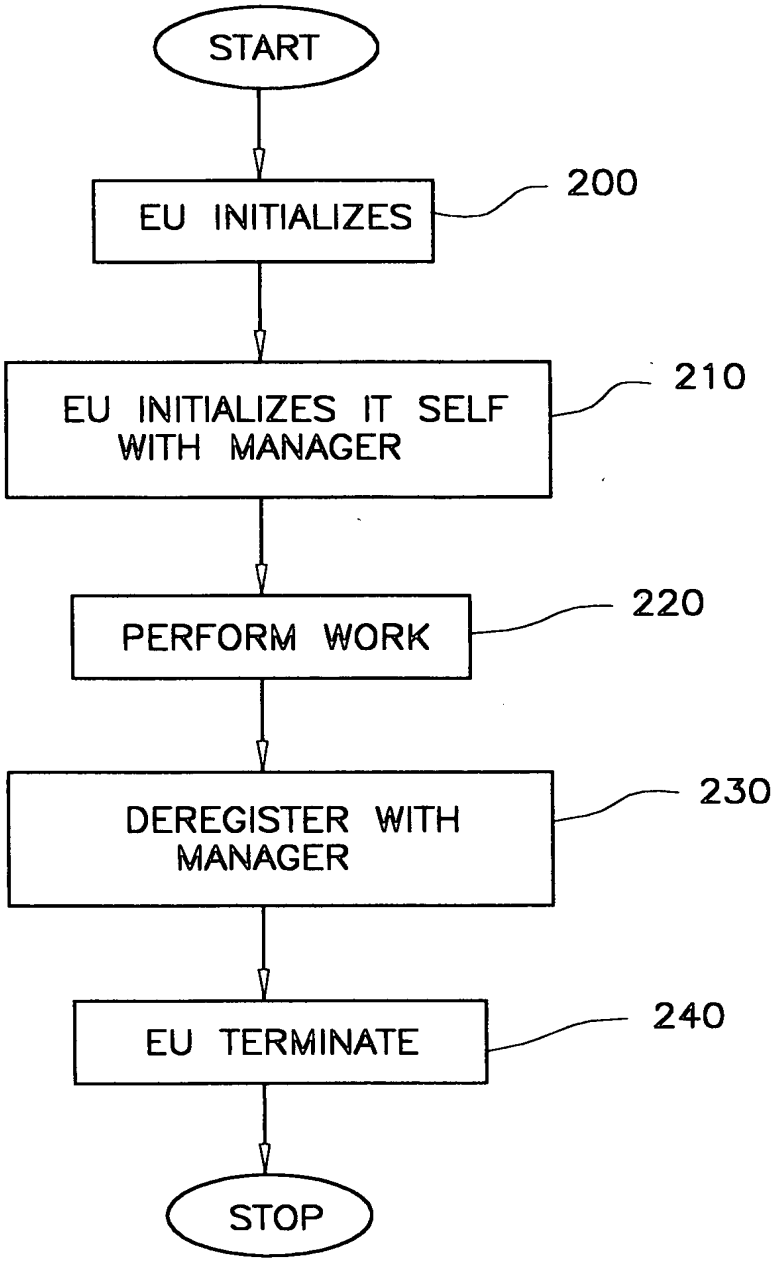
START
EU INITIALIZES
200
EU INITIALIZES IT SELF WITH MANAGER
210
PERFORM WORK
220
DEREGISTER WITH MANAGER
230
EU TERMINATE
240
STOP

FIG. 6

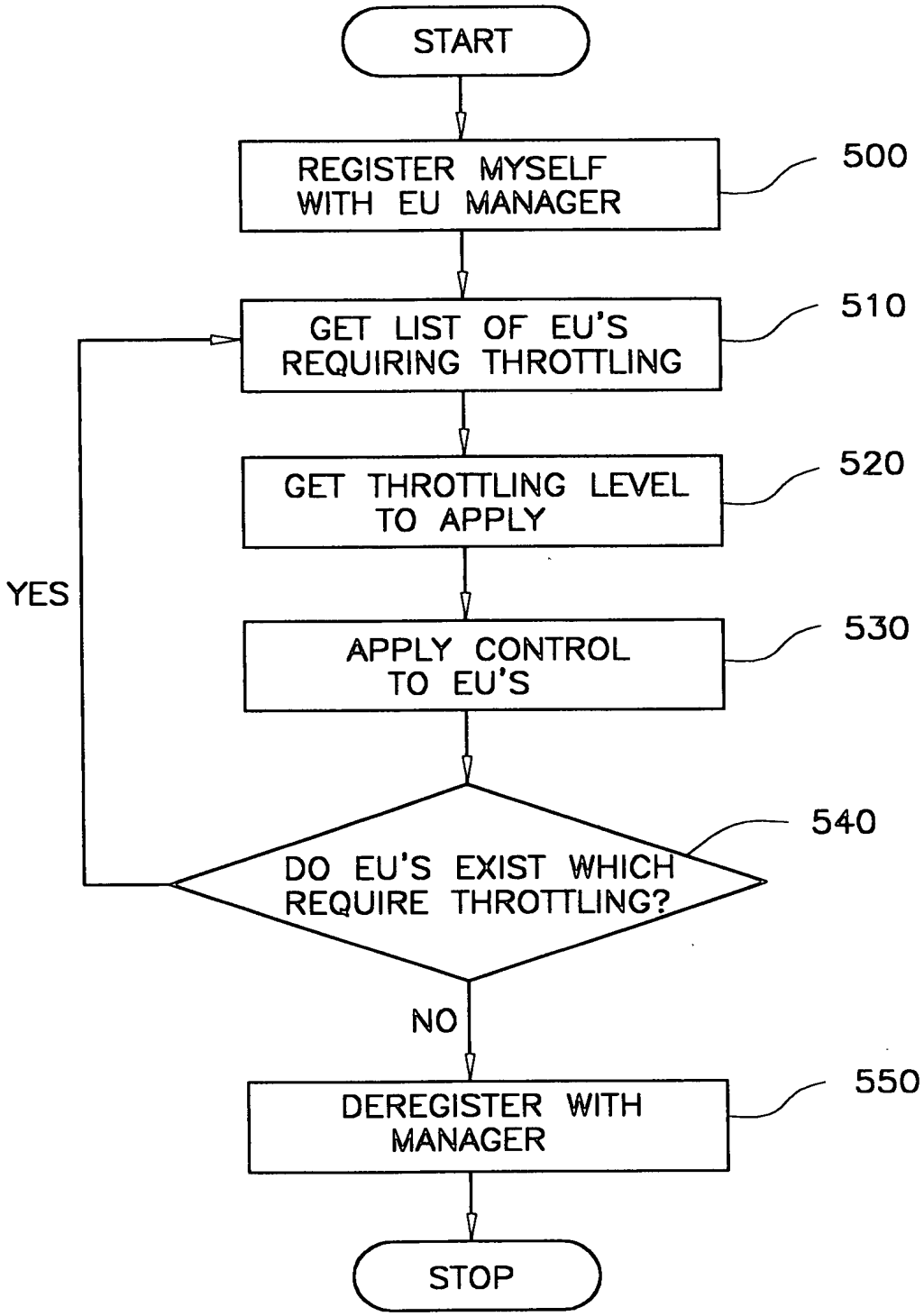
START
REGISTER MYSELF WITH EU MANAGER
500
GET LIST OF EU'S REQUIRING THROTTLING
510
GET THROTTLING LEVEL TO APPLY
520
APPLY CONTROL TO EU'S
530
DO EU'S EXIST WHICH REQUIRE THROTTLING?
540
YES
NO
DEREGISTER WITH MANAGER
550
STOP

FIG. 6B

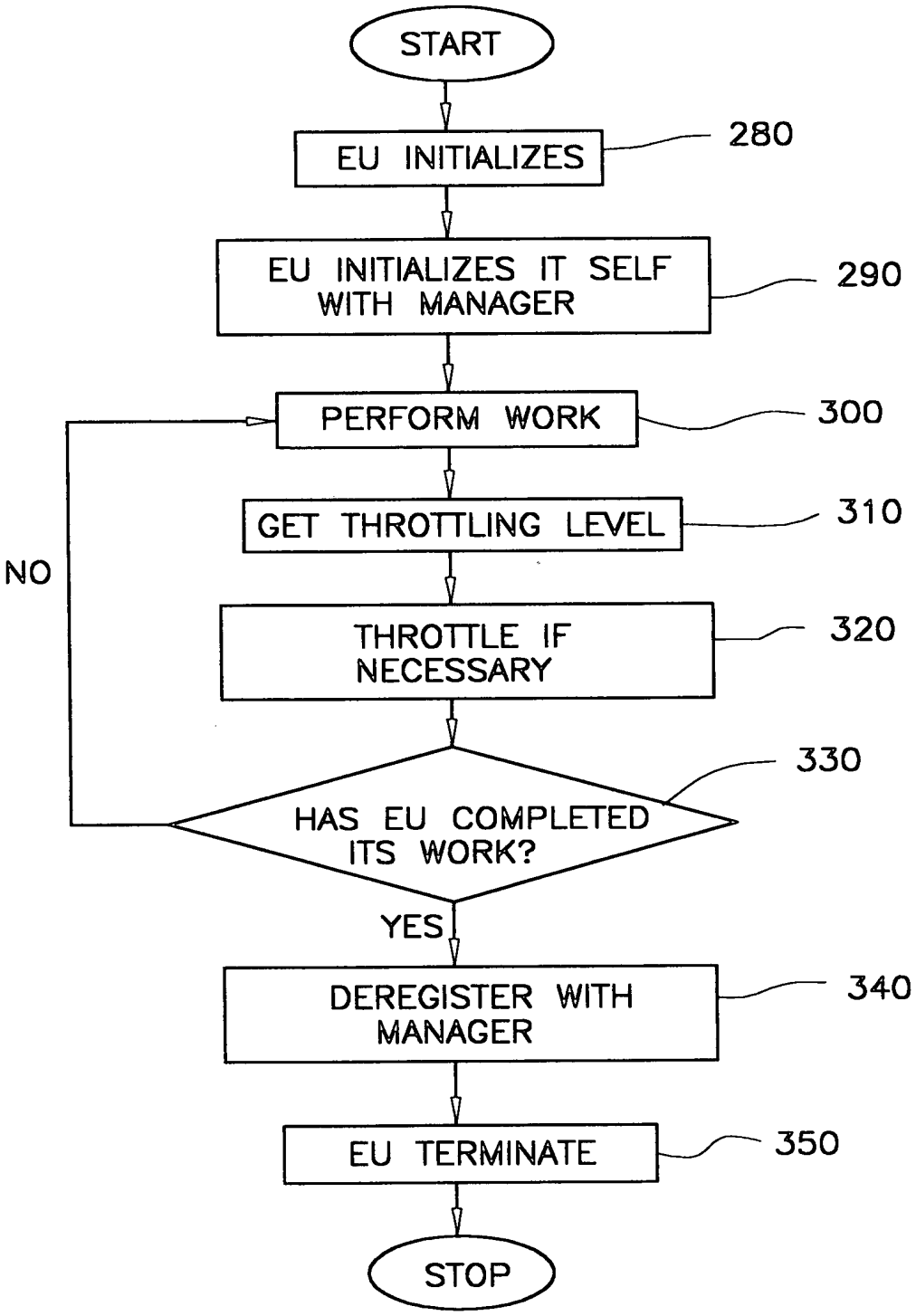
START
EU INITIALIZES
280
EU INITIALIZES IT SELF WITH MANAGER
290
PERFORM WORK
300
GET THROTTLING LEVEL
310
THROTTLE IF NECESSARY
320
HAS EU COMPLETED ITS WORK?
330
NO
YES
DEREGISTER WITH MANAGER
340
EU TERMINATE
350
STOP

FIG. 7

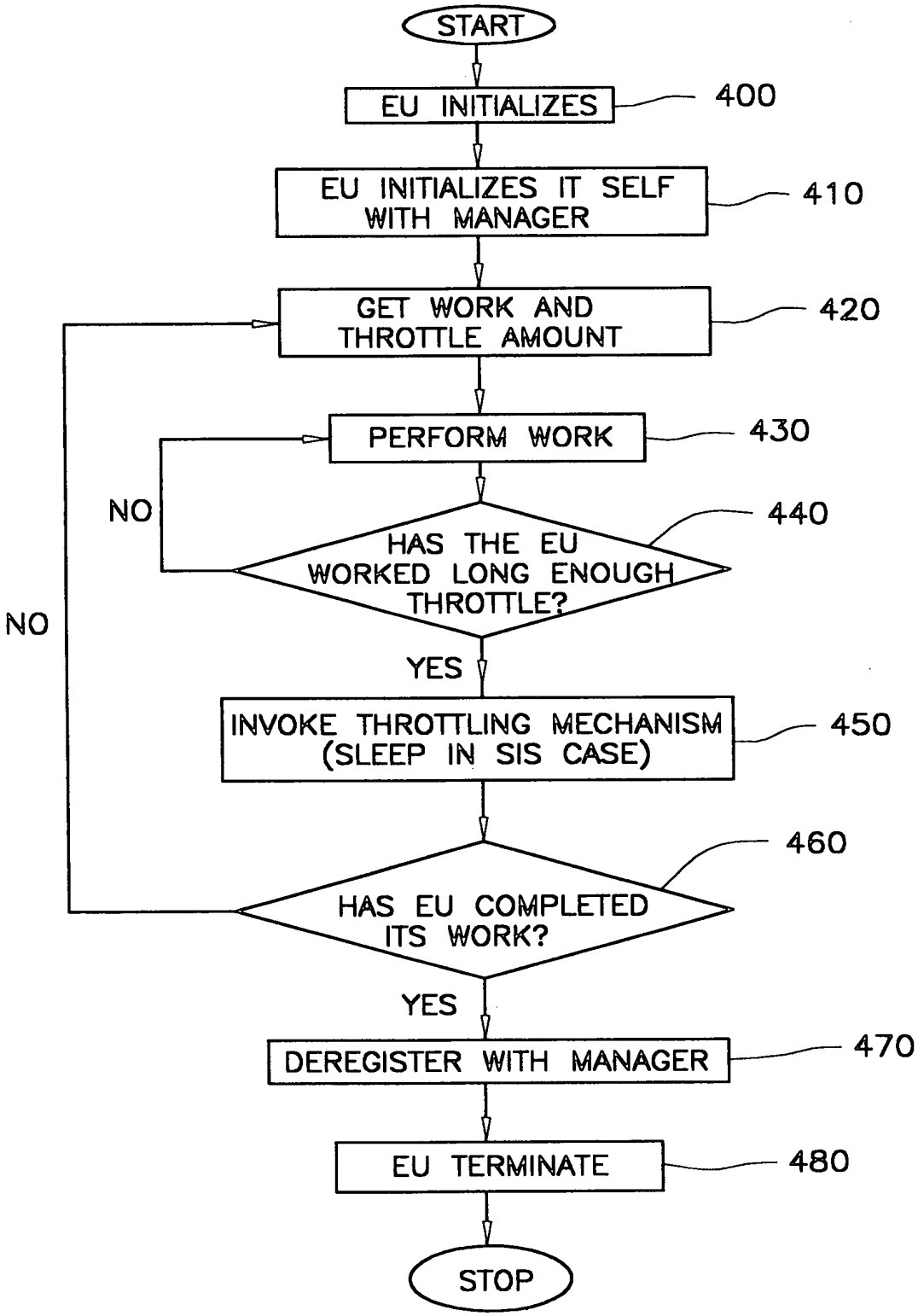
START
EU INITIALIZES
400
EU INITIALIZES IT SELF WITH MANAGER
410
GET WORK AND THROTTLE AMOUNT
420
PERFORM WORK
430
NO
HAS THE EU WORKED LONG ENOUGH THROTTLE?
440
NO
YES
INVOKE THROTTLING MECHANISM (SLEEP IN SIS CASE)
450
HAS EU COMPLETED ITS WORK?
460
YES
DEREGISTER WITH MANAGER
470
EU TERMINATE
480
STOP

FIG. 8

```
COMPONENT ThrottleAgent
BEGIN
EUManager->RegisterThrottleAgent();
While (EUManager->IsThrottlingRequired())
BEGIN
EUList=EUManager->GetListOfEUs(self);
ThrottlingLevels=EUManager->GetThrottlingLevel(EUList);
...apply ThrottlingLevel to EUs In EUList....
END
EUManager->DeregisterThrottleAgent() ;
END
```

FIG. 9

```
COMPONENT EU
BEGIN
...Initialization...
EUManager->RegisterEU(...args...);

While(NOT Done)
BEGIN
...Do some work...
ThrottlingLevel=EUManager->GetThrottlingLevel(self);
Throttle(ThrottlingLevel);
END

...Termination...
EUManager->DeRegisterEU();
END
```

FIG. 10

```
FUNCTION Utility()
BEGIN
      WHILE (NOT done)
      BEGIN
          ...do some work...
          Throttle If Needed()
      END
END
```

(a) Inserting SIS point

```
FUNCTION Throttle If Needed()
BEGIN
      (workTime, sleepTime) = GetThrottlingLevel() ;
      timeWorked = Now() - workStart ;
      IF (timeWorked > workTime)
          SLEEP ( sleepTime ) ;
          workStart = Now() ;
      ENDIF
END
```

(b) SIS implementation

FIG. 11

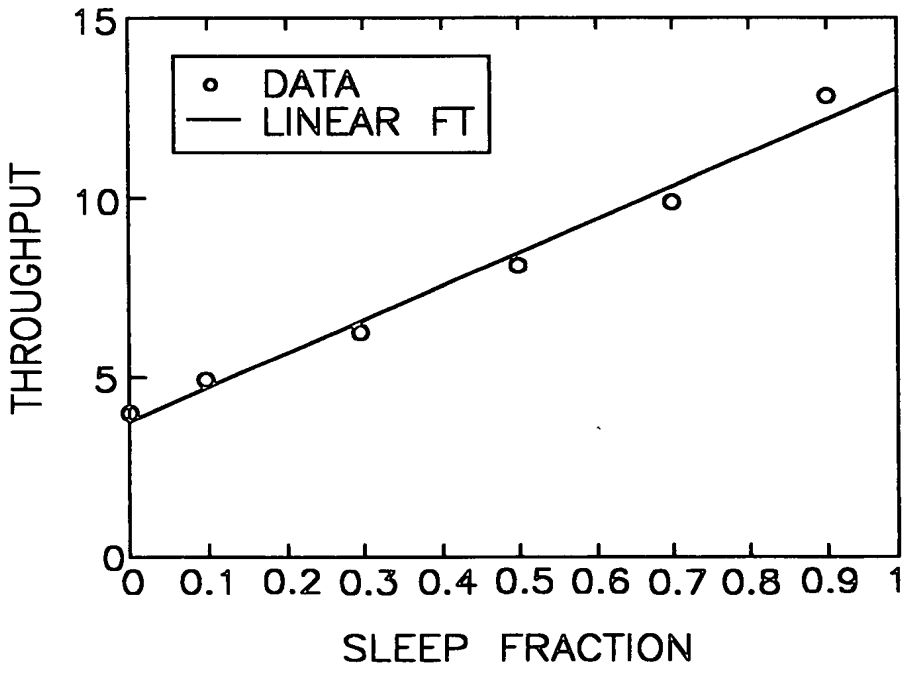
15
10
5
0
THROUGHPUT
DATA
LINEAR FT
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
SLEEP FRACTION

FIG. 12

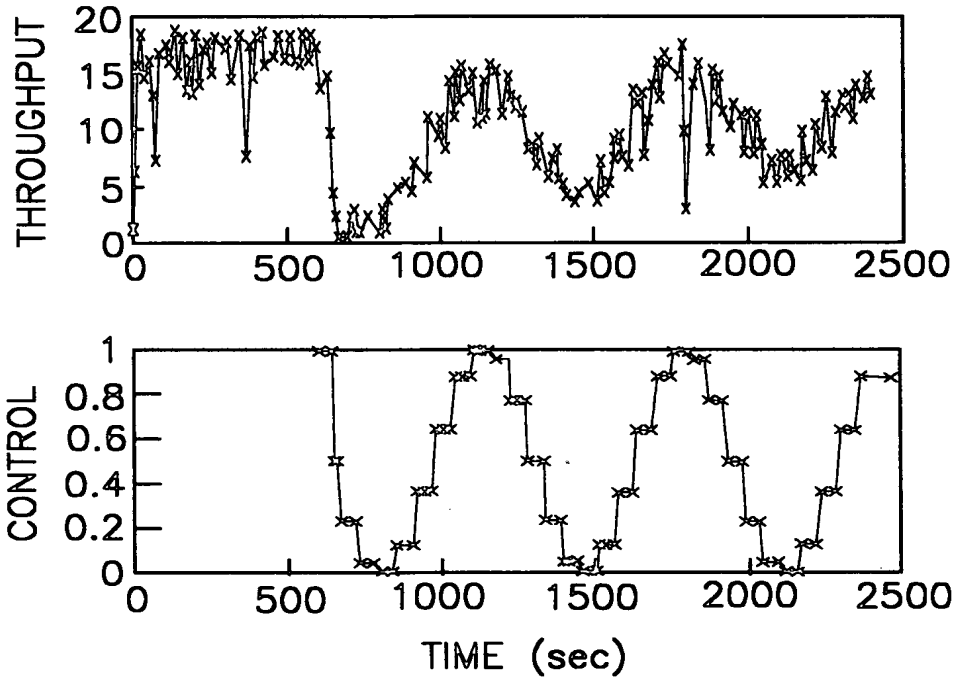
THROUGHPUT
20
15
10
5
0
0
500
1000
1500
2000
2500
CONTROL
1
0.8
0.6
0.4
0.2
0
0
500
1000
1500
2000
2500
TIME (sec)

FIG. 13